AD 641263

ESD-TR-66-499

COMPUTER-CENTERED DATA BASE SYSTEMS

Glenn A Buckles, 1st Lt, USAF
Stewart B Carpenter, 1st Lt, USAF

OCTOBER 1966

DIRECTORATE OF COMPUTERS
DEPUTY FOR ENGINEERING & TECHNOLOGY
ELECTRONIC SYSTEMS DIVISION
AIR FORCE SYSTEMS COMMAND
United States Air Force
L.G. Hanscom Field, Bedford, Massachusetts

Distribution of this document is unlimited.

ESD-TR-66-499

COMPUTER-CENTERED DATA BASE SYSTEMS

Glenn A Buckles, 1st Lt, USAF
Stewart B Carpenter, 1st Lt, USAF

OCTOBER 1966

DIRECTORATE OF COMPUTERS
DEPUTY FOR ENGINEERING & TECHNOLOGY
ELECTRONIC SYSTEMS DIVISION
AIR FORCE SYSTEMS COMMAND
United States Air Force
L.G. Hanscom Field, Bedford, Massachusetts

Distribution of this document is unlimited.

FOREWORD

This research was performed in the Directorate of Computers, Deputy for Engineering and Technology, L. G. Hanscom Field, Bedford, Massachusetts. The Advanced Data Management (ADAM) system was developed by The MITRE Corporation, Bedford, Massachusetts.

Review and Approval:

This technical report has been reviewed and is approved.

CHARLES A. LAUSTRUP
Colonel, USAF
Director of Computers

ABSTRACT

This paper presents a discussion of a specific information system, the Advanced Data Management (ADAM) system, which was established to help the designers and users of large computer-centered data-base systems. The historical background of information systems is included to provide a foundation for discussing the concepts and techniques of ADAM. An application of the ADAM system to a real problem is presented to illustrate the usefulness and practicability of the ADAM design philosophy. The main point which the authors wish to establish is that computer-centered data-base systems as exemplified by ADAM provide a useful and more advanced philosophy concerning data-base systems development than currently exists.

COMPUTER-CENTERED DATA BASE SYSTEMS

Information Systems exist in many forms and are used for many applications. Such systems combine people and computers to provide commanders and their staffs with information about personnel, material, intelligence, and weapons.

This paper presents a discussion of a specific information system, the Advanced Data Management (ADAM) system, which was established to help the designers and users of large computer-centered data-base systems. The historical background of information systems is included to provide a foundation for discussing the concepts and techniques of ADAM. An application of the ADAM system to a real problem is presented to illustrate the usefulness and practicability of the ADAM design philosophy.

The main point which the authors wish to establish is that computer-centered data-base systems as exemplified by ADAM provide a useful and more advanced philosophy concerning data-base systems development than currently exists.

OVERVIEW

Although information systems have been developed by operational users, specialized contractors, and Air Force Systems Command, all development efforts have utilized essentially a direct approach. The eventual user identified his requirement, and the requirement was then translated into detailed design by specialists, the system designers. The detailed design was used in the production of an actual information system usually by a third group of specialists, computer programmers. As the original requirements moved from originator to the specialists, they were subjected to interpretation and change resulting from simple communication problems. Each specialist was called upon to make design decisions which influenced the eventual system's capability. The decisions were often made without knowledge of the requirement. Even if systems went from originator to final producer totally unchanged, the time lag involved often meant that the original requirement had changed. Thus, this direct approach which went from requirements to design to final system produced results which did not totally satisfy the military requirement.

In the development of weapons systems and support equipment, testing in the early design phases and evaluation of prototypes is a standard procedure. Such testing is often omitted for the computer programming sub-systems of information systems. One reason for this failure to test is a lack of adequate laboratory tools. The ADAM system was built by the MITRE Corporation for the Electronic Systems Division, Air Force Systems Command, to act as a test bed for information systems during their conceptual, definition, and design phases. ADAM permits the user and system designer to build a functional prototype of the proposed system in a laboratory environment. Tests can be performed on the prototype to determine if the logical design satisfies the user's actual requirement. In particular, the prototype building and testing provides a means of communication between the user and the system designer. It also permits the identification of elements in the user's requirement which are likely to change with time.

To prove the advantages and usefulness of the ADAM system to the design and development of information systems required its utilization on a real application. A prototype of an information system which will assist in officer personnel assignment was one of the first real applications of the ADAM system. The final information system will be developed utilizing a classic development approach of design, test, evaluate, and redesign rather than the typical direct approach used to build previous information systems. Only by providing the user and system designer with testing tools such as ADAM can the classical approach be utilized.

The Man-Job-Match project, which was responsible for the prototype of the desired information system, was initiated by the Deputy Chief of Staff for Personnel, Hq USAF as a part of a total program associated with officer assignments. Air Force Systems Command through the Electronic Systems Division prepared the prototype system within ADAM. The basic problem was to design and develop a computerized assignment system which meets the requirements and objectives of the long range personnel management systems.

The specific details of the system and the techniques to accomplish them were not fully known when work first began. Therefore, the design for the prototype was not totally defined for the system designer and was subject to modification caused by clarification of personnel policies and objectives by the user.

The user supplied the data, the general system logic, and the operating concepts from which a system was to be designed. The data was the fundamental element of the system. It consisted of information about men, information about jobs, and current personnel assignment policies. Approximately 4.5 million characters of man and job data were used in the development effort.

The general logic description defined the assignment process very superficially without regard to technical requirements. The process was described in four functional steps:

1. Find men available for reassignment and determine vacant jobs.

2. Evaluate every available man's qualifications against every vacant job's requirements.

3. Determine assignment.

4. Transfer assignment results to the man and job files. Application of the appropriate personnel policies to each step furnished sufficient clarification for the system designer to begin his efforts.

The problem which the designer faced was how to develop a design which accomplished the ultimate goal of the project and satisfied the operating concepts of the user. The operating concepts were very general, but posed some very pointed design questions. Four concepts of particular interest were:

1. The prototype system had to be responsive to the manager's needs.

2. The assignments produced by the model had to satisfy the manager's concept of a good assignment.

3. The model had to be developed to fully demonstrate a variety of assignment actions.

4. The model had to be a dynamic system with the capability to simulate complete movement of a representative force over a designated period of time.

Many explicit functions which were implied by these guidelines had to be accommodated.

It was the designer's job to assure that each of the user's requirements were incorporated in a comprehensive, functional model. The basic characteristics which the designer was required to accommodate were:

1. A large data base consisting of approximately 4.5 million characters of information about Air Force men, jobs, and personnel policies.

2. The capability to retrieve specific information from vast amounts of data such as in the determination of available men and vacant jobs.

3. Computational requirements at various points within the system. One type of computation consisted of simple arithmetic operations for determining qualification weights for men versus jobs. Another was a complex optimization process for determining the assignments of men to jobs.

4. An effective data manipulation capability to accomplish large volume updates. Updates were based on the assignments produced by the system and required placement of data into man and job records to reflect projected actions or actual assignment transfers.

5. A report generation facility to satisfy the manager's need for information about the system. Hard copy or cathode-ray-tube displayed reports about men, jobs, assignment results, and system status for off-line or on-line evaluation were imperative.

These five characteristics are those associated with a class of information systems which are called data-base systems. Such systems are also referred to as data management systems or information storage and retrieval systems.

Because the Man-Job-Match project incorporates requirements which correspond to the defining characteristics of data-base systems, it might be considered typical of a class of problems for which data-base systems are well suited. ADAM is a system which contains generalized techniques for accommodating data-base systems characteristics. It provides a new design approach to data-base systems which was applied to the Man-Job-Match effort. A discussion of the application is presented in a later section of the paper.

BACKGROUND

Data-base systems have been used in numerous applications. Because of the lack of design tools, the implementations of these data-base systems have not been accompanied by the testing of the design against the application's requirements during the early stages of development. The ADAM system was built to permit the testing of data-base systems in their design phase, thus allowing a designer to follow a more orderly development approach.

A background discussion which describes both the background of data-base systems and the technological evolution of these systems will provide a context in which to associate ADAM with related efforts. Following the background discussion, the characteristics of ADAM will be presented as well as an actual example of the use of ADAM in the design of a data-base system, the Man-Job-Match prototype.

To associate ADAM with related efforts in data-base systems, a brief sketch of the evolution of such systems during the last few years is needed. Let us begin by examining the range of applications of data-base systems. After a discussion of applications, the effects of technological developments in the computer field on data-base systems will be explored.

DATA BASE SYSTEMS' APPLICATIONS - MILITARY AND BUSINESS

One of the earliest applications of computers was in the area of finance. Today, no one is surprised to find payrolls calculated by computers and to have company books on magnetic tape. Computerized military finance systems have been utilized for years. The applications can all be considered either a special purpose data-base system or utilization of a general purpose data-base system, depending upon how a particular application was implemented on the computer. Calculating a payroll requires:

1. A data base of employees.

2. A capability to retrieve and select employees.

3. Computational routines for determining the actual pay.

4. Data manipulation capability to handle status changes.

5. A report generation facility to present results and prepare the actual checks.

In other words, a payroll system requires the five characteristics of data-base systems. Another early application of computers within both the military and the business world was in the area of logistics. Current literature is filled with talk of totally computerized logistics systems for business. Such systems would continually monitor inventories by initiating purchase requests when inventory levels reach appropriate points. These systems in turn would keep track of projected deliveries and would log the stock into the system upon arrival as well as sending follow-up requests on late deliveries. The reason such systems are discussed as being within the current technology is that computer applications which perform these functions have been operating for several years. The essential characteristics of any of the logistics applications are the same as those of data-base systems. The Requirements Computation System of the Air Force Logistics Command (1) or the Air Force base supply system are excellent examples.

DATA BASE SYSTEMS' APPLICATIONS - MILITARY SUPPORT

The applications of data-base systems have not been limited to those areas which are common to both the military and business, such as finance and logistics. Intelligence data have likewise been handled by data-base systems for several years. (2) An Intelligence Data Handling System (IDHS) (4) exists for both the IBM 1410 and the IBM 7090/7094 computers and is utilized within the Air Force. The Navy has, during the last five years, developed several data-base systems to aid in the handling of intelligence data.

Reliability and maintainability data on weapon systems is handled at the Ballistic Systems Division by a data-base system. A large and sophisticated data-base system is being developed for the Reliability Center at the Air Force Systems Command's Rome Air Development Center. (3) The assignment of airlift aircraft is still another example of a military application of data-base systems. The preceding examples are in the areas of military support applications.

DATA BASE SYSTEMS' APPLICATIONS - MILITARY OPERATIONS

Data-base systems have found application even in the vital area of military operations by aiding the commander in maintaining knowledge of the status of his forces. Major elements of some of the Air Force's largest Command and Control systems are data-base systems. The 473L system for Headquarters United States Air Force has a very sophisticated data-base system which is technically noteworthy of description in the professional literature. (5) The system acquires, processes, and displays data necessary for timely action decisions by Hq USAF and for USAF recommendations to the Joint Chiefs of Staff.

A basic element in the 465L System which transmits, collects, processes, and displays data to assist the Commander, Strategic Air Command, in commanding and controlling his forces is a data-base system. The interim Command and Control System used by the United States STRIKE Command is an on-line data-base system developed by the Electronic Systems Division and the MITRE Corporation. (6,9) Analysis of the Semi-Automatic Ground Environment (SAGE) system, which operates as part of the continental air defense capability, has identified some of the characteristics of data-base systems within the system. These are just four of the examples where data-base systems have helped the commander with his status of forces.

These examples from the operational area as well as those from areas of military support and business show the range of applications of data-base systems. With such a wide range of applications exhibiting similar characteristics, it is no surprise that general purpose techniques for handling these characteristics have been developed. In this sketch of the recent evolution of data-base systems, the development of the techniques for general purpose data-base systems should now be described.

GENERAL PURPOSE DATA BASE SYSTEMS

As the utilization of computers increased, common characteristics of data-base systems began to be recognized. Computer hardware as furnished by the manufacturers came with utility computer programs to

aid in the new applications. The aid was in the form of general techniques based upon the common characteristics of data-base systems. By 1959 a committee of computer manufacturers and users in cooperation with the United States Department of Defense was formed to attempt some standardization and synthesis of the developments in data-base systems.

One of the results produced by the committee was a set of specifications for a Common Business Oriented Language (COBOL). This was one of the first widespread recognitions of the common characteristics of data-base systems. The language essentially provided a mechanism for describing the data base, operating on the data, and specifying the form of the results. Computer manufacturers have since implemented many COBOL compilers. Subsequently, systems designers and computer programmers have used these COBOL compilers for many applications.

As with any standard and technologically dependent effort, time and usage have identified weaknesses and deficiencies. The professional literature is filled with discussions of COBOL's virtues and faults, (7, 8) but these discussions are not germane to the current sketch. All that is necessary to note is that the data-base systems have continued to evolve for such reasons as: (a) ease of use; (b) requirement to handle more complex files; (c) efficiency; and (d) "not build here" attitudes. Because of the continued evolution of data-base systems, the work of the COBOL developers should be viewed as a plateau but not as a termination point. Evolution from this COBOL plateau proceeded along two lines: (a) techniques for improved capability of general purpose data-base systems for the handling of applications; and (b) techniques for improved man and computer communication during both the production of a system to handle an application and the operation of the application.

DATA BASE SYSTEMS WITH IMPROVED CAPABILITY

Both the military and commercial manufacturers of software and hardware have developed general purpose systems with improved capabilities. The Mark III system of Informatics Corporation (9) is an example of a new general purpose data-base system whose objective is easier use. General Electric in their Integrated Data Store (IDS) made a direct extension to COBOL so that random access devices

could be efficiently utilized. (9,10) The BEST system of NCR is still another example of a development by a commercial firm since COBOL was introduced in 1959. (9,11) IBM is currently implementing GIS, a general purpose data-base system, on their new series of computers, the IBM 360's. (12,13)

The military, in particular the operational support activities, has also produced general purpose data-base systems during the last five years. The Intelligence Data Handling Systems (IDHS) for both the IBM 1410 and IBM 7090/7094 evolved during this period. (4) The ABACUS system which assists the Directorate for Studies and Analysis, Deputy Chief of Staff for Plans and Operations, Headquarters USAF is a recently developed general purpose data-base system. (14)

An important characteristic of all general purpose data-base systems is their separation of techniques from applications. These systems have been prepared without a detailed knowledge of either the information in the data base or the operations and calculations to be performed. Instead of using such knowledge, the systems were designed to accept as inputs the details of the application -- the data base and the process to be performed. There has always been some separation of the techniques (the computer programs) and the application (the data). In the general purpose systems this separation is greater than those found in the older data-base systems prepared for only one specific application.

Another characteristic of general purpose data-base systems is that they function in the conventional mode of operation for computer programs, namely off-line. An application or problem is analyzed and the inputs to the computer are prepared. These inputs are then forwarded to the computer facility where they are inserted into the machine and results are produced. These results are then returned and studied. Because the entire process is lengthy, the languages of general purpose systems have tended to allow many options for the results. Usually, the rules for selecting the options are rigid and somewhat cumbersome to apply, so a detailed knowledge of the language is required. Systems which are designed for non-programmers actually are systems which are easy for non-programmers to use. This ease of learning and use is another characteristic of the systems previously described. Although they do require some effort to utilize, they are certainly easier to use than assembly (machine) level programming.

Although off-line operation is the standard mode with the current second generation computers, the announced third generation will make on-line operations more common. The man/machine interface is more important and vital in such on-line operations. Recognizing this important problem area, several general purpose data-base systems which address the man/machine interface have been developed in the research community. The techniques used within such systems constitute the other line of evolution from the COBOL plateau.

The most extreme example of a data-base system which emphasizes the man/machine interface is the Advanced Experiment Sturcture for On-Line Planning (AESOP) developed by the MITRE Corporation. (15) An outstanding characteristic of AESOP is a display system which helps the user select the data he desires. The language used to select specific information is shown to him on the display scope and the system provides him with cues and choices as he forms a statement in the language. Operations and calculations upon the data may also be develcped on the display scope through use of an associated typewriter. During the entire operation, the user is time-sharing the computer while he prepares and operates his applications.

Another example of an on-line and time-shared data-base system is the LUCID system and its follow-on, the Time Shared Data Management System (TDMS), both developed by Systems Development Corporation. (16, 17) The outstanding characteristic of LUCID is a query language which is very easy to learn and use. With such a language the user can quickly select his data and then operate upon the data outside of the system. Error messages and other systems aids provide a user with learning tools.

A third on-line system is the Compile On-Line and GO (COLINGO) system developed by the MITRE Corporation (6, 18). Versions of this system operate on IBM 1410 and 1401's at United States STRIKE Command, National Range Division, and the MITRE Corporation. Although the system is not time-shared, a single user is provided on-line access through a console typewriter. The language used to operate upon the data base is somewhat complex but still user-oriented. An interesting characteristic of the system is the ability to store statements and processes for later operation. This storing capability is the characteristic

of many time-sharing systems which gives them an appeal. A user can build and test a complex process on-line, then save it. When the process is required, all that is needed is to select it and add the parameters. COLINGO provides a language for describing processes on data bases while on-line.

The preceding sketch has discussed the application and technological evolution of data-base systems during the last few years. The sole objective of such a discussion was to provide a frame of reference for a description of ADAM, its objectives and characteristics.

ADAM, A DESIGN TOOL

There are two major topics in any discussion of the ADAM system. First, functional descriptions of the characteristics of ADAM are required. Along with such functional descriptions, the techniques which are used to achieve the capability are described. The combination of the functional elements of ADAM into a prototype of a real application completes an ADAM discussion. For this discussion, the prototype and application to be considered is officer personnel assignment. But before discussing this application in detail, the first topic, the functional description of ADAM's characteristics, must be covered.

Two distinct categories of users utilize ADAM. First, there is the user with an operational requirement. The other is the system designer who wishes to utilize ADAM to build a prototype of a data-base system which will satisfy a specific requirement. This second category of users will be called the user/designer. Each type of user views the ADAM system differently. To the operational user, the ADAM system and his prototype look like one large specific data-base system which satisfies his requirements. To a user/designer, the distinction between the ADAM system and the prototype is much more definite. ADAM is the tool the user/designer had when he began to build. The prototype is all the data and special programs which the user/designer had to prepare. In describing the functional capabilities of ADAM, it is better to take the view of a user/designer.

During any description of ADAM, an astute observer will notice that similarities between ADAM and general purpose data-base systems.

Such similarities exist because of a basic design philosophy; namely, the separation of the details of a problem (the data) from its solution (the computer programs). This philosophy is necessary in both ADAM and general purpose systems because detailed knowledge of the application was not available when the computer programs which comprised the system were prepared. ADAM differs from general purpose data-base systems in that it incorporates another basic design philosophy -- an ability to accept modification at all levels of design. It is this second philosophy which makes ADAM a laboratory tool and not a production system. However, aside from lack of efficiency and practical limitations, nothing prevents any user from utilizing an ADAM prototype in a production operation. The ability to accept modification within a general purpose design is the major asset of ADAM for a user/designer.

DATA BASE GENERATION

In considering how ADAM functionally accepts large data bases, the first basic characteristic of data-base systems, both the general purpose and modifiable features will be discussed. To introduce a data base into the ADAM system and, hence, into the computer, its structure and data types must be described in a file description language. Provided within ADAM is a table driven translator which permits the definition of any file description language. An Initial File Description Language is also available within the system. Thus, a user/designer can elect to develop, insert, and test his own file description language using ADAM or he may describe his data base using the existing language. If he elects to remain with the Initial File Description Language, the user/designer has modification options such as special conversions for specific data elements when they are inserted into ADAM.

The user/designer can also specify special conversion and handling routines for specific data elements when they are retrieved and stored. For example, in retrieving information on radar returns, a user/designer could specify that special tables are used to locate the most current information. Of the four real applications of ADAM, the user/designer has always elected to use the Initial File Description Language with special conversion routines rather than prepare a specialized language.

Another feature available when a user/designer inserts the data base is the automatic coding of data. By designating data as having a certain property, dictionaries are formed which associate a number with each unique value of the data. These dictionaries are automatically used when the data are retrieved so that the user receives the same values that were inserted. The user/designer has the preceding features available within ADAM to allow him to handle a large variety of data bases.

DATA SELECTION AND RETRIEVAL

The selection and retrieval of data, the second characteristic of a data-base system, is handled by most systems through a query language. Because the process of selection and retrieval is repeated more often and by more users than the process of initially inserting the data into the system, query languages need to be more user oriented and are more susceptible to specific requirements. As previously noted, current systems tend to operate in an off-line fashion while future systems will be on-line operations, or both. The ADAM system has been designed to operate (a) off-line through tape inputs, (b) on-line through display scopes, typewriters, and printers; or (c) mixed with off-line inputs sending results on-line and on-line inputs initiating complex off-line processes.

Besides this ability to operate in any combination of modes, the user/designer also can develop his own query language. As in data-base description, the user/designer can either use the table - driven translator to specify his own query languages or utilize a query language, FABLE, which is provided with the ADAM system.

Regardless of his choice, the user/designer still has another modification feature available; string substitution. This feature permits a user/designer to equate a single word to a string of words. When a query enters the system it is examined for these single words and when one is found the equivalent string of words is substituted. After each substitution the string can optionally be examined again for further substitutions.

An additional design feature is available for those user/designers concerned with developing on-line user interaction through a display scope, a typewriter, or both. The display capability of ADAM allows

the user/designer to assign meaning to different display actions, for example, depressing buttons or light-gun selection of an item which is displayed on a display scope. These assigned meanings can be associated with each other so that they can form a query which selects and retrieves data. This capability of specifying sets of display actions which can select data will be very useful for the user/designer who will be concerned with the on-line data-base systems of the future.

COMPUTATIONS

Adam provides the user/designer with two basic capabilities for handling computation requirements, the third characteristic of data-base systems. The first capability is through a query language. As in the selection and retrieval of data, a user/designer can either develop his own query language for computation or use the FABLE language provided with the ADAM system. A difficulty with using a query language is that the entire computational process must be presented in a single statement. Thus, logical testing of the data with computation performed on the test result or computations based upon results previously calculated are not easily described in query languages. For extensive calculation, a higher-order algebraic language is desirable and is the other basic capability for handling computations. ADAM provides the user/designer with the ability to insert a FORTRAN program into the system to handle extensive computations. Such a program operates under a special monitor within the ADAM system and access to the data base is through special routines. Since the data that the FORTRAN program uses comes from the data base, standard input/output statements of FORTRAN are not allowed. Instead, data and results must be passed between the data base by means of special routines usually prepared by the user/designer. The FORTRAN program can then be initiated by a statement in query language.

DATA MANIPULATIONS

The query language is also the way to handle the fourth characteristic of data-base systems -- many data manipulations such as in mass updates. The previous comments about query languages applies to this area. In particular, efficiency may require that such operations occur off-line and/or be done by special purpose routines.

REPORTS GENERATION

The final characteristic of data-base systems, reports generation, is also handled by the ADAM system. The user/designer must prepare his formats off-line. ADAM is an on-line system only when it is acting as a prototype. While the user/designer is building his prototype he is usually working off-line. The report generator of ADAM performs similarly to most report generators. A format for a report is prepared which is independent of the actual data to be presented in the finished report. The report generator takes both the format and the data, combines them, and adjusts the resulting report for output to any of the devices of the ADAM system; i.e., printers, display scopes, teletypes, or typewriters. Provisions have been made in the language which describes the formats to allow specia routines to be executed during the report generation process. This capability helps the user/designer prepare formats which present the data and results in ways which are most convenient for the user.

During the previous discussion, the functional characteristics of ADAM have been briefly described. Some of the choices and capabilities provided a user/designer have been indicated. To demonstrate how a user/designer combines these capabilities, a real application should be considered. The prototype of the Man-Job-Match project is just such an application.

MAN-JOB-MATCH: AN ADAM APPLICATION

The initial problem that faced the Man-Job-Match project was the data base. The two files which constitute the major portion of the data base were not large or complex by operational standards, but they presented several difficulties when viewed from a research and development aspect. The total data base consisted of approximately 4.5 million characters of data: 5400 man records of 594 characters and 5400 job records of 318 characters.

The general nature of the data was important to the decisions regarding data base handling. There were properties which contained numeric codes, alpha-numeric codes, dates which required special conversion, and strings of alphabetic information. It was necessary to be able to accommodate these various data and, at the same time, to facilitate the operations which were to be applied to them.

The Initial File Description Language of the ADAM system was considered more than adequate for the file definition of the assignment system. This language provided for introducing the data types present in the data base:

1. Logical property descriptions for internal coding of alpha-numeric codes and data strings (e.g. duty location, education specialty, special experience).

2. Numeric property descriptions for storing numeric values (e.g. tour length, social security number, job number).

3. Raw value property descriptions for alphabetic character strings (e.g. job titles, names of men).

4. An option for using user-generated conversion routines (e.g. to convert the dates which appeared in various forms).

Each of the property types except the raw value property were represented in the computer in a form which allows querying.

PERSONNEL POLICY

Introduction of the Man and Job files into ADAM satisfied most of the operational data base requirements of the assignment model. The personnel policy data presented a different problem for the user/designer. The difficulty was one of versatility rather than size. Policy data were to be used as parameters to the model. User-defined guidelines indicated a variable structure of policy data and emphasized that the data could change frequently. Repetitious groups of similar data were evident in the policy data. There was no way to predetermine the number of repetitions that might exist within one policy structure. Policy data were easily accommodated by the Initial File Description Language. The solution to the problem of frequent changes to the policy files was found in ADAM's FABLE query language.

AVAILABILITY SELECTION

Interrogation and retrieval of data was the second area of concern for the user/designer. System requirements for data retrieval were extensive.

The determination of available men and vacant jobs was a selective data retrieval action. The selection process was governed by specific requirements expressed in qualifying (boolean) statements which were often complex. The criteria for identifying available officers for reassignment consisted of at least five attributes:

1. career field and specialty

2. current duty location by major air command area

3. rank

4. date of arrival at current duty station and availability date

5. projected assignment actions

It was necessary to form queries which would screen the entire file of men and retrieve specific data about those who met the criteria. Essentially the same process was necessary to obtain vacant jobs.

FABLE was selected as the language in which to implement the data retrieval messages. There was sufficient versatility in the language to describe the qualifying phrases to the detail necessary for the model's successful implementation. The use of string substitutions was extensive throughout the model. These string substitutions, as described in a preceding section, permitted the user/designer to equate a single word to a string of words. When the query entered the system all these single words were replaced by their corresponding string of words. The use of substitution made the job of the personnel manager easier as he exercised the model. To exercise a portion of the model, the personnel manager needed to type only a few words rather than a long FABLE statement. The availability query, which consists of 15 characters when the personnel manager entered it, expanded to approximately a 1300 character statement.

CRITERIA MODIFICATION

The lengthy qualifying expression used in the queries had to be modified as criteria changed. One change to an expression required that the entire expression be restructured. String substitutions allowed the user to segment criteria into individual relationships and assign a name

to each component. Under this arrangement, a component (a word and its corresponding string of words) requiring modification could be deleted and redefined without affecting the rest of the expression. For example, in the case of determining available men, each attribute was described using string substitutions, "GET MAN AVAILABLE" was the query entered by the manager. The word "AVAILABLE" was the name of the highest level of qualifying expressions. The string of words which was substituted for the word AVAILABLE contained words which required further substitution. These words AFSC, GRADE, MACAREA, DATEAR, and PROJECTED were names of the specific criteria for the respective attributes of career specialty, grade, location by major command area, date of arrival on station, and projected actions. Such string substitutions contributed greatly to satisfying the requirement of responsiveness to the personel manager's needs. The personnel manager needed only to change the string of words corresponding to GRADE to change the criteria of qualification for grade. To add an entirely new criterion required only changing the string for AVAILABLE. Such usage of string substitution permitted the personnel manager to easily exercise the model and to easily change criteria.

COMPUTATIONAL REQUIREMENTS

Computational requirements existed in two of the functional modules of the prototype. Each requirement differed greatly in mathematical complexity and method of data handling. These areas of difference identified additional capabilities required to implement the model--(1) a means for facilitating mathematical routines of varying complexity and (2) a flexible data manipulation capability.

A special monitor in ADAM (COMFORT) accommodated the mathematical routines. However, since data input and output functions were excluded from the monitor, the problem of supplying data to the routines had to be handled by specially prepared programs.

The computational requirements were defined as: (1) qualifying men versus jobs, and (2) optimizing the assignment of men to jobs. The first of the two was defined as a two step process. The first step checked to determine if a man meets the mandatory job requirements as defined by personnel policyand, if he did, the second step evaluated his characteristics versus the requirements of each vacant job. This procedure necessitated data from four sources:

1. a file of available men

2. a file of vacant jobs

3. a policy file of mandatory requirements

4. a policy file of qualification data

Data generated by the qualification procedure was placed into a fifth file. The optimization routine, which was a mathematical optimization process, required the data in matrix form. The resulting assignments and associated data were placed into still another file.

No adequate capability existed within ADAM to handle the I/O functions associated with the computational routines; therefore, it was necessary to write specific assembly language programs to accomplish the task. These programs were entered into the ADAM system in the same manner as the mathematical routines.

The computational processes were very special cases of data manipulation used in the assignment model. A more conventional application of data manipulation existed in the posting of assignments and updating of the data base. Assignments made by the assignment model were placed in an ADAM file. These assignments had to be reflected in the records of the appropriate men and jobs. A cross-file reference and update was necessary to accomplish changes to the man and job files.

ASSIGNMENT POSTING AND UPDATING

FABLE was selected to accomplish the tasks of posting and updating. It let the user/designer formulate queries to do the necessary cross-file referencing and subsequent posting of data in a single step. The posting was based on the assignment results. Each job to which a man had been assigned had to be located in the master file of jobs, similarly, each man had to be located in the master file of men. Appropriate data (e.g. man identification, projected date of assignment, etc.) was stored from the man record into the job record. Similarly, job data (e.g. job number, projected date of assignment, etc.) was placed in the man's

record. All these operations were required to record projected assignments in the master files. The updating function consisted of advancing the time frame of the model, checking the projected date of assignment of each record in the man and job files, and if appropriate, transferring projected assignment data into "current assignment" data fields of the new records. For assignments of large numbers of men and jobs each function involved large amounts of data and required retrieval of data from the rather large master files.

DATA REPORTS

The final product of the assignment prototype was information which had to be transmitted at some point to the user. Since the model was an experimental tool, more information was required from the system by the user than just assignment results. In some cases the form and timeliness of the information often were important to the user. For example, special formats were prepared to report the assignment results together with man and job data from the master files. Because ADAM's capability for formatting output was flexible, but somewhat difficult to use efficiently, the standard formats generated by the ADAM system were used to report most data on the model.

The timeliness of the data required by the user was an important aspect of the experimental prototype. Recall from earlier discussions that data-base systems can have two computing modes associated with them -- on-line and off-line. Both modes were utilized in the development of the assignment model. The elements of the system were developed and checked out in the off-line mode. The use of the model then, in an on-line mode, allowed an effective and comprehensive solution to the timeliness of the data the user sees. In addition, on-line operation made the user an integral part of the model at experimentation time.

MAN/MACHINE INTERACTION IN MAN-JOB-MATCH

To conclude the discussion of the assignment model development, a short explanation of the man/machine interaction features of the model is appropriate. The model is structured in functional components each having assignment parameters or information the user might desire to manipulate. In the on-line mode, the user can enter messages by typewriter, push-button, and/or light-gun actions. The push buttons and light gun are input devices associated with a cathode ray tube; i.e., a display scope. He may receive information on the printer, typewriter,

or cathode ray tube. The model is supplied with a set of queries associated with push-buttons which are considered to be standard for the system, regardless of policy changes. To initiate an assignment cycle, the user, who in this case is a personnel manager or specialist, executes the stored queries for obtaining available men and determining vacant jobs. He may at this time examine the results of these queries by displaying the subsetted files on one of the output devices. He then may choose to alter availability or vacancy criteria by changing the appropriate string substitutions which comprise the availability or vacancy queries. If so, he must delete the current subset of men and jobs and again execute the queries, which now include his modifications. Normally, the altering of string substitutions is done by typewriter.

The manager can now initiate the qualification process. This is initiated by push-button action. The policy files, both for mandatory requirements and qualification criteria, are created prior to the assignment run. The manager may choose to alter the policy files at any time by light-gun actions associated with push-button activated queries, or through typewriter-entered messages. The results of the qualification procedure can be reviewed just as the available men and vacant jobs were.

The assignment process may be initiated by push-button action and is subsequent to the qualification process. The assignment results can be displayed by the manager for his evaluation. Hard copy and cathode ray tube (CRT) displays are generally requested. The manager can make general comments about an abbreviated CRT display immediately and may desire to examine the results and associated data about men and jobs more closely. At this point, the manager can make a decision about the assignment results which will determine his next step. He may accept the model produced assignments. If he does, he initiates the posting and updating queries which are associated with push-buttons. The updating will move the time frame on the model so that the next cycle of the model will select different men and jobs. If he does not accept the assignments, he may elect to do any or all of the several options allowed him with regard to personnel policy:

1. change availability or vacancy criteria to alter the subsets

2. change mandatory or qualification policy

3. arbitrarily select a group of men and jobs for assignment action

After selecting the appropriate option or updating, the personnel manager can exercise the entire model again. This interaction with the basic elements of design at every point in the assignment process permits the personnel manager to use the prototype to design his final system.

Appendix I is Phase I of a script used in demonstrating the basic assignment model as an on-line prototype. The events covered in Phase I are those necessary to accomplish an assignment cycle and to display selected data associated with those events. Phase I, in conjunction with subsequent phases which demonstrate the capability to incorporate changes to the model, shows how a personnel manager can exercise the model in accordance with the dictates of changing policy.

It is important at this point to emphasize that the man/machine interaction supplies the capability to accomplish the final three stages of the design/test/evaluate/redesign cycle. In the Man-Job-Match problem the testing and evaluation take the form of a structured experiment. A specialist in the personnel assignment field institutes policies which he deems appropriate and causes assignments to be made. His professional opinion, which is supported by experience in making personnel assignments, is the basis for evaluating the results. Changes to the prototype, whether to the policies or routines, will effect a change in the final system's design. Subsequent repetitions of this cycle will evolve the design to a status which is appropriate for implementing within an operational environment.

CONCLUSION

By using ADAM to build a prototype of an officer assignment system, the personnel manager and the system designer have a test bed on which to verify their design. Such a prototype provides two important functions:

1. a means of communication of requirements between an operational user and a system designer

2. a mechanism for identifying the elements of a design which are most frequently changed.

The Man-Job-Match project has completed the building of a prototype within the ADAM system. Extensive testing of the prototype utilizing personnel specialists and real data began in the summer of 1966. Even when this testing leads to a final design, such a design still must be translated into an operational system. Such an operational system will be one of the first data-base systems developed along the classic approach of design, test, evaluate, and redesign. The key to such an approach for data-base systems is the availability of proper test vehicles such as ADAM. The effectiveness of ADAM as a design tool has been tested with the Man-Job-Match project. When the prototype built in ADAM has led to the design of a successful operational system, then the users and designers of future information systems will have both a proven new design tool and a different development approach.

APPENDIX I

MAN-JOB-MATCH DEMONSTRATION SCRIPT

MODEL I

Phase I - Basic Model Assigning Lieutenants.

ITEM/PUSH BUTTON	RESULTING ACTION/REMARKS

1. Enter via typewriter: LET SHOW MEAN (D1 D2). This allocates the display units to be used in the demonstration

2. 3, 11	Deletes the current subset of the MAN file, (MANAVAIL file) from the model.
3. 3, 12	Subsets the MAN file in accordance with the criteria for determining the availability of men for reassignment. This subset becomes the new MANAVAIL file.
4. 3, 13	Displays the MANAVAIL file.
5. 4, 11	Deletes the current subset of the JOB file (JOBAVAIL file) from the model.
6. 4, 12	Subsets the JOB file in accordance with the criteria for determining the job vacancies. This subset becomes the new JOBAVAIL file.
7. 4, 13	Displays the JOBAVAIL file.
8. 5, 11	Initiates the qualification of the men for jobs: a. Checking on mandatory job requirements. b. Scoring the men on desired qualifications.
9. 5, 12	Displays the results of the qualification routine. (JMWEIGHT File)

APPENDIX I CONT'D

ITEM/PUSH BUTTON		RESULTING ACTION/REMARKS
10.	6, 11	Assigns the qualified men to jobs.
11.	6, 12	Displays the final assignment matrix.
12.	7, 11	Posts the assignments to the MAN and JOB files.
13.	7, 12	Displays the posted results from the MAN file.
14.	7, 13	Displays the posted results from the JOB file.
15.	8, 11	Updates the dates in the Man and JOB files.
16.	8, 12	Displays the posted dates from the JOB file.
17.	8, 13	Displays the posted dates from the MAN file.

REFERENCES

1. Foreman, A. C., "Advanced Data Management Experiment", IEEE Transactions on Aerospace and Electronic Systems, Vol AES-2, No. 1, (January 1966), 115,120.

2. "Advanced Programming Developments, A Survey", Directorate of Computers, Electronic Systems Division, U. S. Air Force, Bedford, Massachusetts, ESD-TR-65-171 (February 1965), 101

3. "Design of Reliability Central Data Management Subsystem" Information Processing Branch, Rome Air Development Center, U. S. Air Force, Griffiss AFB, New York, Vol. 2, RADC-TR-65-189, (July 1965)

4. "Intelligence Data Handling Systems (IDHS) Users Manual", Rome Air Development Center, U. S. Air Force, Griffiss AFB, New York.

5. "Operational Test Capability (OTC) Query Language", 473L, Electronic Systems Division, U. S. Air Force, Bedford, Massachusetts, ESD-TDR-64-443, (June 1964)

6. Spitzer, J., J. Robertson, and D. H. Neuse, "The COLINGO System Design Philosophy". Proceedings of the 2nd Contress of the Information Systems Sciences, Spartan Books, (November 22, 1964)

7. Bromberg, Howard, "What COBOL Isn't", Datamation, Vol. 7, No. 9 (September 1961) 27-29

8. Communications of the ACM, Vol. 5, No. 5, (May 1925)

9. Proceeding of the Second Symposium on Computer-Centered Data Base Systems, System Development Corporation, Santa Monica, California, TM-2624/100/00, (December 1965)

10. "Introduction to Integrated Data Store", General Electric Computer Department, CPB-1048, (April 1965) 50 pages

11. Ziegler, James R., "Computer-Generated Coding", Datamation, Vol. 10, No. 10, (October 1964) 59-61

12. "Generalized Information System-Application Description" International Business Machines Corporation, E20-0179-0, (1965) 42 pages

13. "Generalized Information System-Briefing Outline", GIS Field Support Group, Data Processing Division International Business Machines Corporation, Bethesda, Maryland, January 1966

14. "ABACUS Users' Manual", Study Information Group, Directorate of Studies and Analysis, Deputy Chief of Staff/Plans and Operation, Hq USAF, Washington, D. C. SIO-66-175-1

15. Bennett, Edward; Haines, Edward; Summers, John K., "AESOP: A Prototype for On-Line User Control of Organizational Data Storage Retrieval and Processing", Proceeding of AFIPS 1965 Fall Joint Computer Conference, Spartan Books, November 1965, 435-455

16. Franks, E. W., "A Data Management System for Time-Shared File Processing Using a Cross-Index File and Self-Defining Entries", Proceedings of AFIPS 1966 Spring Joint Computer Conference, Spartan Books, April 1966, 79-86

17. Franks, E. W., "A Data Management System for Time-Shared File-Processing Using a Cross-Index File and Self-Defining Entries", Systems Development Corporation, Santa Monica, California, SP-2248, April 1966

18. "COLINGO-C10 Users' Manual", MITRE Corporation, Bedford, Mass, MTR-35, 1966

19. Connors, Thomas B., "ADAM- A Generalized Data Management System", Proceedings of AFIPS 1966 Spring Joint Computer Conference, Spartan Books, April 1966, 193-204

NOTE: An Air Force Film is available (SFP1433, ADAM - A Computer Programming System) from the Air Force Film Library that depicts the concept of ADAM.

Security Classification

DOCUMENT CONTROL DATA - R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1 ORIGINATING ACTIVITY (Corporate author) Directorate of Computers Deputy for Engineering & Technology Electronic Systems Division, L.G. Hanscom Field, Mass.	2a REPORT SECURITY CLASSIFICATION Unclassified 2b GROUP N/A

3 REPORT TITLE

COMPUTER-CENTERED DATA BASE SYSTEMS

4 DESCRIPTIVE NOTES (Type of report and inclusive dates)

5 AUTHOR(S) (Last name, first name, initial)

Buckles, Glenn A. 1st Lt, USAF
Carpenter, Stewart B. 1st Lt, USAF

6 REPORT DATE	7a TOTAL NO OF PAGES	7b NO OF REFS
October 1966	28	19

8a CONTRACT OR GRANT NO In-House b PROJECT NO c d	9a ORIGINATOR'S REPORT NUMBER(S) ESD-TR-66-499 9b OTHER REPORT NO(S) (Any other numbers that may be assigned this report)

10 AVAILABILITY/LIMITATION NOTICES

Distribution of this document is unlimited.

11 SUPPLEMENTARY NOTES	12 SPONSORING MILITARY ACTIVITY
None	Electronic Systems Division L.G. Hanscom Field, Bedford, Mass.

13 ABSTRACT

This paper presents a discussion of a specific information system, the Advanced Data Management (ADAM) system, which was established to help the designers and users of large computer-centered data-base systems. The historical background of information systems is included to provide a foundation for discussing the concepts and techniques of ADAM. An application of the ADAM system to a real problem is presented to illustrate the usefulness and practicability of the ADAM design philosophy. The main point which the authors wish to establish is that computer-centered data-base systems as exemplified by ADAM provide a useful and more advanced philosophy concerning data-base systems development than currently exists.

DD FORM 1 JAN 64 1473

Security Classification

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT

INSTRUCTIONS

1. ORIGINATING ACTIVITY: Enter the name and address of the contractor, subcontractor, grantee, Department of Defense activity or other organization *(corporate author)* issuing the report.

2a. REPORT SECURITY CLASSIFICATION: Enter the overall security classification of the report. Indicate whether "Restricted Data" is included. Marking is to be in accordance with appropriate security regulations.

2b. GROUP: Automatic downgrading is specified in DoD Directive 5200.10 and Armed Forces Industrial Manual. Enter the group number. Also, when applicable, show that optional markings have been used for Group 3 and Group 4 as authorized.

3. REPORT TITLE: Enter the complete report title in all capital letters. Titles in all cases should be unclassified. If a meaningful title cannot be selected without classification, show title classification in all capitals in parenthesis immediately following the title.

4. DESCRIPTIVE NOTES: If appropriate, enter the type of report, e.g., interim, progress, summary, annual, or final. Give the inclusive dates when a specific reporting period is covered.

5. AUTHOR(S): Enter the name(s) of author(s) as shown on or in the report. Enter last name, first name, middle initial. If military, show rank and branch of service. The name of the principal author is an absolute minimum requirement.

6. REPORT DATE: Enter the date of the report as day, month, year; or month, year. If more than one date appears on the report, use date of publication.

7a. TOTAL NUMBER OF PAGES: The total page count should follow normal pagination procedures, i.e., enter the number of pages containing information.

7b. NUMBER OF REFERENCES: Enter the total number of references cited in the report.

8a. CONTRACT OR GRANT NUMBER: If appropriate, enter the applicable number of the contract or grant under which the report was written.

8b, 8c, & 8d. PROJECT NUMBER: Enter the appropriate military department identification, such as project number, subproject number, system numbers, task number, etc.

9a. ORIGINATOR'S REPORT NUMBER(S): Enter the official report number by which the document will be identified and controlled by the originating activity. This number must be unique to this report.

9b. OTHER REPORT NUMBER(S): If the report has been assigned any other report numbers *(either by the originator or by the sponsor)*, also enter this number(s).

10. AVAILABILITY LIMITATION NOTICES: Enter any limitations on further dissemination of the report, other than those imposed by security classification, using standard statements such as:

(1) "Qualified requesters may obtain copies of this report from DDC."

(2) "Foreign announcement and dissemination of this report by DDC is not authorized."

(3) "U. S. Government agencies may obtain copies of this report directly from DDC. Other qualified DDC users shall request through ______________________."

(4) "U. S. military agencies may obtain copies of this report directly from DDC. Other qualified users shall request through ______________________."

(5) "All distribution of this report is controlled. Qualified DDC users shall request through ______________________."

If the report has been furnished to the Office of Technical Services, Department of Commerce, for sale to the public, indicate this fact and enter the price, if known.

11. SUPPLEMENTARY NOTES: Use for additional explanatory notes.

12. SPONSORING MILITARY ACTIVITY: Enter the name of the departmental project office or laboratory sponsoring *(paying for)* the research and development. Include address.

13. ABSTRACT: Enter an abstract giving a brief and factual summary of the document indicative of the report, even though it may also appear elsewhere in the body of the technical report. If additional space is required, a continuation sheet shall be attached.

It is highly desirable that the abstract of classified reports be unclassified. Each paragraph of the abstract shall end with an indication of the military security classification of the information in the paragraph, represented as *(TS)*, *(S)*, *(C)*, or *(U)*.

There is no limitation on the length of the abstract. However, the suggested length is from 150 to 225 words.

14. KEY WORDS: Key words are technically meaningful terms or short phrases that characterize a report and may be used as index entries for cataloging the report. Key words must be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location, may be used as key words but will be followed by an indication of technical context. The assignment of links, rules, and weights is optional.